ASSIGNMENT-1

[ASM'20]

Assignment: 1 (Mainstream and Bioinformatics)

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Geeral instructions:

Regarding your file:

1) Submit **only running** code that you have tested before.
2) Use the provided template to write your code. Do not use any other templates.
3) Submit **only** the content of **main.asm** file. You have to press "Submit" button in the form.
4) Submit your code through this Google form link

5) Do **not** use for ALL questions:
    a. Conditional directives, such as .if, .while, etc …
    b. MUL or SHIFT or ROTATE instructions
6) Use only instructions taken up until Lab 7.
7) Do not hard-code answers. It is clear that it will be considered unacceptable behavior.

This assignment is intended for *individual* contribution. Sharing ideas or part of answers is considered plagiarism and will not be tolerated (= ZERO mark for the entire assignment). All submissions will be checked for plagiarism automatically.

Marks will be **deducted** for not following any of the above submission rules (-5/instruction).

PLEASE WATCH THE AUTOGRADER EXPLANATION VIDEO CAREFULLY BEFORE SUBMITTING YOUR CODE

Notes:

1. All the sample runs were taken from the assignment's template.
2. Array values are not read in one line. They are read line-by-line (i.e., separated by new lines).

Good luck
Assembly team 2020

Assignment: 1 (Mainstream and Bioinformatics)

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Q1. In the procedure called "Q1", that rotates a list by k elements. For example [1,2,3,4,5,6] rotated by two becomes [3,4,5,6,1,2]. Solve this **without** using **shift** or **rotate** instructions.

Sample Input:

Enter number of Elements: 6

Enter your Elements: 1

2

3

4

5

6

Sample Output:

Your Rotated Elements are: [3, 4, 5, 6, 1, 2]

Sample Run:

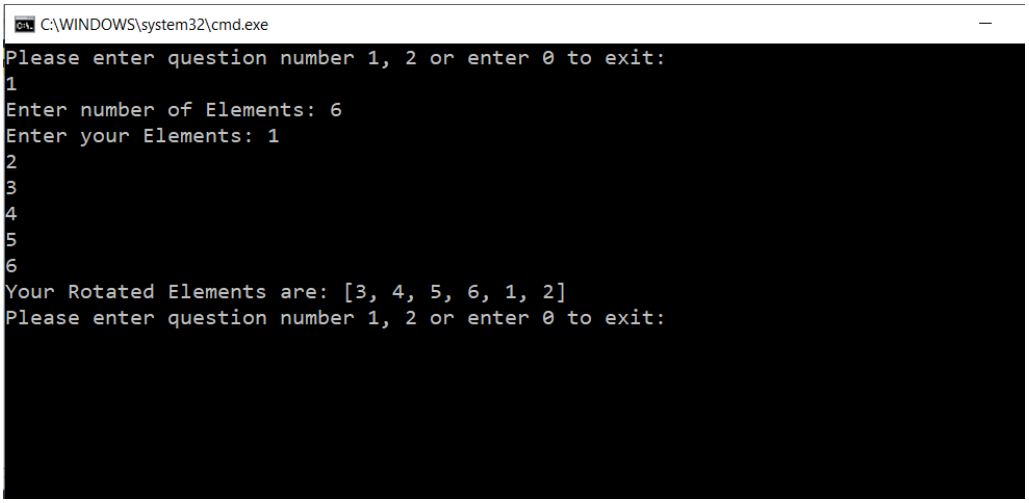

Figure 1 question-1 sample run

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Q2. In the procedure called "Q2", write assembly code that swaps two characters in a word. The program should read the indices of the two characters to swap and the word. The program should display the word again with the characters swapped.

Constraints:

- The string will not exceed 30 characters.

Sample Input:

Please, Enter your first index: 4

Please, Enter your second index: 2

Please, Enter your word: Hello

Sample Output:

Your Swapped word: Hlleo

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
2
Please, Enter your first index: 4
Please, Enter your second index: 2
Please, Enter your word: Hello
Your Swapped word: Hlleo
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 2 question-2 sample run

Q3. In the procedure called "Q3", write assembly code that reads two words from the user: the first one is a correct spelling of a word, and the second one is a correct/incorrect spelling of the first word. Determine if the second word is spelled correctly. Finally, output the second word and the degree of correctness.

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

The degree of correctness is as follows:

- "CORRECT" if it is an exact match
- "ALMOST CORRECT" if no more than 2 letters are wrong
- "WRONG" if 3 or more letters are wrong

Constraints:

- Words will contain only upper-case letters.
- The maximum word length is 20.
- Each word will be on a separate line.

Sample Input:

SAMPLE

SIMPLE

THEIR

THEIR

WINDMILL

WINDOWS

ASSEMBLY

ASSEMBLYY

Sample Output:

SIMPLE IS ALMOST CORRECT

THEIR IS CORRECT

WINDOWS IS WRONG

ASSEMBLYY IS ALMOST CORRECT

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
3
SAMPLE
SIMPLE
SIMPLE IS ALMOST CORRECT
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
3
THEIR
THEIR
THEIR IS CORRECT
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
3
WINDMILL
WINDOWS
WINDOWS IS WRONG
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
3
ASSEMBLY
ASSEMBLYY
ASSEMBLYY IS ALMOST CORRECT
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 3 question-3 sample run

Q4. In the procedure called "Q1", write assembly code that reads 3 inputs from the user:

1) A number N
2) N intervals, where each interval consists of 2 numbers, a start and end number
3) A number M

Find the number of intervals that the input number M exists in.

Constraints:

- $1 \leq N \leq 10$

Good luck
Assembly team 2020

Assignment: 1 (Mainstream and Bioinformatics)

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Sample Input *(comments are written for illustration and are not included in the sample run):*

Enter the number of intervals N: 5 ; N

Enter the N intervals:

1 ; Interval 1

10

5 ; Interval 2

10

15 ; Interval 3

25

7 ; Interval 4

12

20 ; Interval 5

25

7 ; M

Sample Output:

The input number exists the following number of times in the N intervals: 3

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Sample Run:

```
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
4
Enter the number of intervals N: 5
Enter the N intervals:
1
10
5
10
15
25
7
12
20
25
7
The input number exists the following number of times in the N intervals: 3
Please enter question number 1, 2, 3, 4, 5, 6 or enter 0 to exit:
```

Figure 4 question-4 sample run

Q5. In the procedure called "Q5", write an assembly code that reads from user positive integer N > 2 then it prints the factorial of that N.

Factorial of an integer N is calculated as follows:

N != N *(N-1) * (N-2)…..*1

Constraints:

- Do not use **MUL or Shift** instructions.

Sample Input:

Please Enter N: 7

Sample Output:

The factorial is: 5040

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Sample Run:

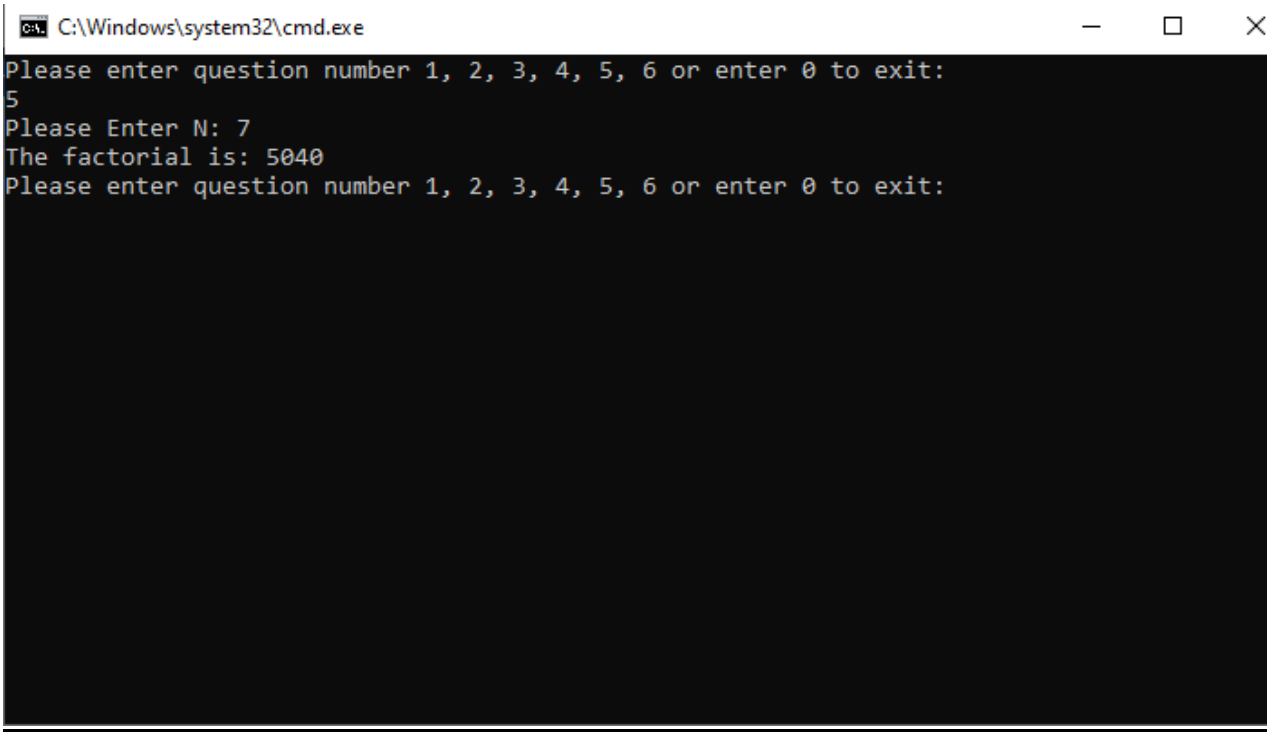

Figure 5 question-5 sample run

Q6. In the procedure called "Q6", write assembly code to find a geometric progression, A geometric progression is a sequence of numbers where each term after the first is found by multiplying the previous one by a fixed, non-zero number called the common ratio.

It is defined by given first term (a), common ratio (r) and (n) number of values in Progression series.

The general form of the geometric progression is:

a, ar, ar^2, ar^3…

For example: 2, 6, 18, 54 - is a **geometric progression** with **r** is 3 and **a** is 2 and **n** is 4.

Your program should read a, r, and n from the user, and displays the geometric progression.

Good luck
Assembly team 2020

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Constraints:

- Do not use **MUL or Shift or Rotate** instructions.

Sample Input:

Enter a: 2

Enter r: 4

Enter n: 5

Sample Output:

The generated sequence:

2

8

32

128

512

Sample Run:

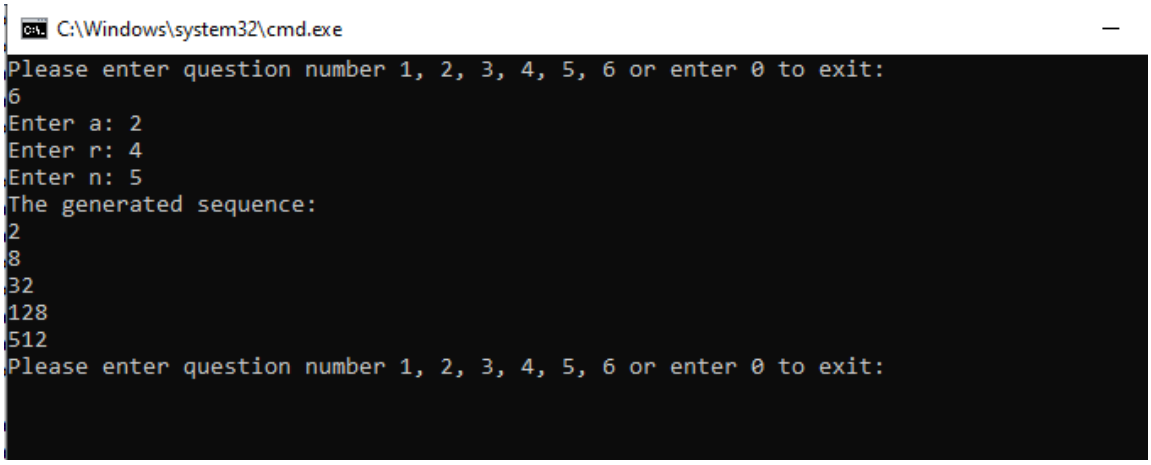

Figure 6 question-6 sample run

Good luck
Assembly team 2020

Assignment: 1 (Mainstream and Bioinformatics)

Faculty of Computer & Information Sciences
Due date: 28 December 2020 Midnight
Course: Assembly language CSW-353
Instructor: Dr. Karim Emara – Dr. Salsabil Amin

Hints:

- WriteInt is an Irvine function that prints an integer value that must be stored in EAX register (number's sign is printed).
- ReadInt is an Irvine function that reads an integer from the keyboard and stores it in EAX register (the input integer is signed).
- WriteDec is an Irvine function that prints an integer value that must be stored in EAX register (number's sign is not printed).
- ReadDec is an Irvine function that reads an integer from the keyboard and stores it in EAX register (the input integer is not signed).
- WriteChar is an Irvine function that prints a character that must be stored in AL register.
- ReadString is an Irvine function that reads a string from the keyboard, stopping when the user presses the Enter key. Pass the offset of a buffer in EDX and set ECX to the maximum number of characters the user can enter. The procedure returns the count of the number of characters typed by the user in EAX.
- WriteString is an Irvine function that writes a string to the console. Pass the offset of a buffer in EDX.
- ReadHex used to read a hexadecimal value from the user. The value after the read is stored in EAX register.
- WriteHex used to write a hexadecimal value to the screen. The value to be displayed is stored in EAX register before calling this procedure.
- ReadChar is used to read a char from the console. The value read from the console is placed in "al" register.
- More about these functions and similar ones can be found in section 5.3 of the book.

Good luck
Assembly team 2020